

In the Claims:

Please cancel Claims 7-9, 17-19 and 27-29; amend Claims 1, 3, 6, 10-11, 13-14, 16, 20-21, 23-24, 26 and 30; and add new Claims 31-39, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented or canceled claims in a continuing or future application.

1. (Currently Amended) A system for loading software applications in an application server, comprising:

an application server for executing ~~[[an]]~~ a software application thereupon, wherein said software application has a ~~number~~ plurality of modules and classes associated therewith;

an application control configuration file associated with said software application, wherein said ~~control~~ configuration file ~~specifies a hierarchy of classloaders to be used with said modules~~ includes a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server;

a deployment ~~mechanism~~ logic that ~~loads with said software application a selection of said classloaders according to the hierarchy specified by said control file~~ parses the configuration file, recognizes the modules and classes specified therein, and retrieves those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file; and ~~[[,]]~~

wherein upon receiving a request to ~~deploy any of said modules~~ load the modules and classes of the software application, the system ~~determines, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the software application, and then deploys those modules~~ constructs an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader.

2. (Original) The system of claim 1 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

3. (Currently Amended) The system of claim 1 wherein said modules are any of [[EJB]] Enterprise Java Bean components, classes, or implementations.

4. (Original) The system of claim 2 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an software application name and a module associated with the application.

5. (Original) The system of claim 4 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

6. (Currently Amended) The system of claim 5 wherein the module is any of [[EJB]] Enterprise Java Bean components, class, or implementations.

7-9. (Canceled).

10. (Currently Amended) The system of claim 1 wherein the server provides multiple [[EJB]] Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

11. (Currently Amended) A method for loading application components, comprising the steps of:

providing an application server with a ~~an EJB~~ software application thereupon, wherein said [[EJB]] software application has a ~~number~~ plurality of modules and classes associated therewith; associating an application configuration file with said software application, wherein said configuration file includes a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server;

~~parsing a control file associated with said EJB software application, wherein said control file specifies a hierarchy of classloaders to be used with said modules~~ the configuration file, recognizing the modules and classes specified therein, and retrieving those modules and classes

from a computer readable medium in a manner consistent with the tag layout in the configuration file;

~~deploying said EJB software application with a selection of said classloaders according to the hierarchy specified by said control file; and [[,]]~~

wherein upon receiving a request to ~~deploy any of said modules~~ load the modules and classes of the software application, ~~determining, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the EJB software application, and then deploys those modules~~ constructing an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader.

12. (Original) The method of claim 11 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

13. (Currently Amended) The method of claim 11 wherein said modules are any of [[EJB]] Enterprise Java Bean components, classes, or implementations.

14. (Currently Amended) The method of claim 12 wherein said user interface allows the software developer to specify a redeploy command that instructs the method that said subset should be redeployed, wherein said redeploy command specifies an [[EJB]] Enterprise Java Bean software application name and a module associated with the application.

15. (Original) The method of claim 14 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

16. (Currently Amended) The method of claim 15 wherein the module is any of [[EJB]] Enterprise Java Bean components, class, or implementations.

17-19. (Canceled).

20. (Currently Amended) The method of claim 11 wherein the server provides multiple [[EJB]] Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

21. (Currently Amended) A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing an application server with a ~~an EJB~~ software application thereupon, wherein said [[EJB]] software application has a ~~number~~ plurality of modules and classes associated therewith; associating an application configuration file with said software application, wherein said configuration file includes a tag layout and application class-loader structure elements that determine the hierarchy of modules and classes of the software application to be loaded into the application server;

~~parsing a control file associated with said EJB software application, wherein said control file specifies a hierarchy of classloaders to be used with said modules~~ the configuration file, recognizing the modules and classes specified therein, and retrieving those modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file;

~~deploying said EJB software application with a selection of said classloaders according to the hierarchy specified by said control file; and [[,]]~~

~~wherein upon receiving a request to deploy any of said modules~~ load the modules and classes of the software application, determining, according to said hierarchy, the minimum number of modules and/or additional modules necessary to deploy or redeploy the EJB software application, and then deploys those modules constructing an application container at the application server with the classes and modules, in the order in which the classes and modules were retrieved, to create a hierarchical class loader.

22. (Original) The computer readable medium of claim 21 further comprising a user interface that allows a software developer to specify a subset of said modules to be deployed.

23. (Currently Amended) The computer readable medium of claim 21 wherein said modules are any of [[EJB]] Enterprise Java Bean components, classes, or implementations.

24. (Currently Amended) The computer readable medium of claim 22 wherein said user interface allows the software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an [[EJB]] Enterprise Java Bean software application name and a module associated with the application.

25. (Original) The computer readable medium of claim 24 wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

26. (Currently Amended) The computer readable medium of claim 25 wherein the module is any of [[EJB]] Enterprise Java Bean components, class, or implementations.

27-29. (Canceled).

30. (Currently Amended) The computer readable medium of claim 21 wherein the server provides multiple [[EJB]] Enterprise Java Bean software applications, each with their own hierarchy of classloaders.

31. (New) The system of claim 1 wherein the application class-loader structure allows for nesting of one or more application class-loader structure elements, and wherein the outermost element of the application class loader structure indicates the application class-loader.

32. (New) The system of claim 1 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an enterprise software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

33. (New) The system of claim 1 wherein the application server supports multiple software applications, each with their own hierarchy of application classloader-structure elements.

34. (New) The method of claim 11 wherein the application class-loader structure allows for nesting of one or more application class-loader structure elements, and wherein the outermost element of the application class loader structure indicates the application class-loader.

35. (New) The method of claim 11 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an enterprise software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

36. (New) The method of claim 11 wherein the application server supports multiple software applications, each with their own hierarchy of application classloader-structure elements.

37. (New) The computer readable medium of claim 21 wherein the application class-loader structure allows for nesting of one or more application class-loader structure elements, and wherein the outermost element of the application class loader structure indicates the application class-loader.

38. (New) The computer readable medium of claim 21 wherein the system allows a software developer to specify a redeploy command that instructs the system that said subset should be redeployed, wherein said redeploy command specifies an enterprise software application name and a module associated with the application, and wherein said redeploy command includes a list of modules relative to the root of the application to be deployed, for redeployment of said modules.

39. (New) The computer readable medium of claim 21 wherein the application server supports multiple software applications, each with their own hierarchy of application classloader-structure elements.